ATTACHMENT C
Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently amended) A method for convolutive encoding and transmission by packets of a digital serial data series flow which is formed by a successive number of bits of a specified value with a current bit of rank k being designated as i(k) ~~of a specified value~~, through a convolutive encoding of depth K, said method comprising the steps:

a) discriminating, in said flow, a set of successive current bits in order to form a current packet of digital data;

b) defining for said convolutive encoding a stable starting binary value;

c) subjecting said digital data ~~i(k)~~ of ~~the~~ said current packet to a convolutive encoding process, at each value of the current bit i(k) corresponding to a first a(k) encoded value and a second b(k) value, ~~the~~ a set of these first and second encoded values constituting an encoded symbol S(k)={a(k); b(k)} representative of the [considered] current bit i(k);

d) forming from the encoded symbols, a packet of encoded symbols, by a ~~packet of encoded symbols by~~ concatenation of said encoded symbols;

e) assigning to said convolutive encoding process said ~~aforementioned~~ stable starting binary value as a constraint value at the end of said packet~~.~~ ;

f) generating at least one encapsulation message of said packet of encoded symbols;

g) transmitting, in ~~the same~~ a common message, said at least one encapsulation message and said packet of encoded symbols for decoding and use; and

h) repeating the operations a) to g) for each current packet of digital data constituting said flow of bits.

2. (Currently amended) A method for decoding a digital data series flow encoded by convolutive encoding using a stable starting binary value and transmitted by packets, said method comprising the following steps:

a) receiving a packet of encoded symbols S(k) and an encapsulation message;

b) discriminating said encapsulation message in order to generate an envelope logic signal having a first binary value prior to the start, and subsequently at the end, of said packet of encoded symbols and a second binary value during the reception of said packet of encoded symbols;

c) subjecting said envelope logic signal and said encoded symbols to a logic processing enabling to generate;

successive pause symbols of specified value and rank on the basis of said stable starting binary value of ~~the~~ a received packet, for said first binary value of said envelope logic signal, and

successive validated encoded symbols S'k={a'(k);b'(k)} of rank k corresponding to said symbols S(k) for said secondary binary value of said envelope logic signal; and

subjecting said successive ~~validated~~ pause symbols and validated encoded symbols of given rank to a continuous VITERBI type decoding, said pause symbols providing ~~enabling the~~ continuity of ~~the~~ a decoding lattice ~~to be obtained~~ by imposing a stable state between two ~~packet~~ packets of successive encoded symbols.

3. (Currently amended) A decoding method according to claim 2, wherein for a pause state as well as for a first logic value of said envelope logic signal corresponding to ~~the signaling of the packet~~ start of a packet and end of a packet equal to the value zero and for a second logic value of said envelope logic signal equal to the value 1, said logic processing comprises applying to each received symbol a(k), b(k) and AND logic function with said envelope logic signal.

4. (Currently amended) A decoding method according to claim 2, wherein for a pause state as well as for a first logic value of said envelope logic signal corresponding to ~~the~~

~~signaling of the packet~~ start of a packet and end of a packet equal to the value 1 and for a second logic value of said envelope logic signal equal to the value 0, said logic processing comprises applying to each received symbol a(k), b(k) an OR logic function with said complemented envelope logic signal.

5. (Currently amended) A decoding method according to claim 2, wherein for a pause state and a first logic value of said envelope logic signal equal to the value 0 and for a signal representing ~~the~~ activity of the envelope corresponding to ~~the~~ a second logic value equal to the value 1, said logic processing comprises applying to each symbol a(k), b(k) an AND logic function with said complemented envelope logic signal.

6. (Currently amended) A decoding method according to claim 2, wherein for a pause state and a first logic value of said envelope logic signal equal to the value 1, and for a signal representing the activity of the envelope corresponding to ~~the~~ a second logic value equal to 0, said logic processing comprises applying to each symbol a(k), b(k) an OR function with said envelope logic signal.

7. (Currently amended) A decoding device for a series flow of digital data encoded by convolutive encoding using a stale starting binary value and transmitted by packets comprising:

means for receiving a packet of encoded symbols S(k) and an encapsulation message;

means for discriminating ~~means of~~ said encapsulation message so as to enable ~~enabling~~ an envelope logic signal to be generated for said current packet, said envelope logic signal having a first ~~logic~~ binary value before the beginning and after the end of the encoded symbol packet and a second ~~logic~~ binary value during the receiving of the packet;

logic processing means for generating:

successive pause symbols of specified value and rank, on the basis of said stable starting binary value of the received packet, for said first binary value of said envelope logic signal, and

successive validated encoded symbols S'k={a'(k); b'(k)} of rank k corresponding to said symbols S(k) for said second binary value of said envelope logic signal; and

VITERBI decoding means for carrying out, on the basis of said pause symbols and said validated encoded symbols of given rank, a continuous VITERBI type decoding, said pause symbols providing ~~enabling the~~ continuity of ~~the~~ a decoding lattice ~~to be obtained~~ by imposing a stable state between two packets of successive encoded symbols.

8. (Currently amended) A device according to claim 7, wherein said logic processing means include at least:

a first logic cell receiving, in a first input, each first symbol element a(k), and, in a second input, said envelope logic signal and delivering a first validated symbol element ~~s'~~a'(k);

a second logic cell receiving, in a first input, each second symbol element b(k), and, in a second input, said envelope logic signal and delivering a second validated symbol element b'(k).

9. (Currently amended) a device according to claim 8, wherein said first and second logic cells are constituted by an identical logic gate, each provided with a first input and a second input, the first input of each logic gate constituting the first input of each logic cell and the second input of each logic gate being connected to the second input of each logic cell directly or by means of a logic inverter, as a function of the logic value assigned to the first and second binary value of said envelope logic signal respectively.

10. (Currently amended) A method for convolutive encoding and transmission by packets of a digital data series flow which is formed by successive number of bits of specified value with a current bit of rank k being designated as i(k) ~~of specified value~~, through a convolutive encoding of depth K, said method comprising the following steps:

a) discriminating in said series flow a set of successive current bits, in order to form a current packet of digital data;

b) defining for said convolutive encoding a stable starting binary value;

c) subjecting said digital data ~~i(k)~~ of ~~the~~ said current packet to a convolutive encoding process, at each value of the current bit i(k) corresponding to a first a(k) encoded value and a second b(k) encoded value, ~~the~~ a set of the ~~these~~ first and second encoded values constituting an encoded symbol S(k)=a{(k); b(k)} representative of the ~~considered~~ current bit i(k) and a plurality of encoded symbols being generated;

d) forming, from the encoded symbols, a packet of encoded symbols by concatenation of said encoded symbols;

e) assigning to said convolutive encoding process said ~~aforementioned~~ stable starting binary value as a constraint value at the end of said packet;

f) generating at least one encapsulation message of said packet of encoded symbols, said message comprising a packet length field;

g) transmitting, in a common ~~the same~~ message, said at least one encapsulation message and said packet of encoded symbols for decoding and use; and

h) repeating the operations a) to g) for each current packet of digital data constituting said flow of bits.

11. (Currently amended) A method for decoding a digital data series flow encoded by convolutive decoding using a stable starting binary value and transmitted by packets, comprising the following steps:

a) receiving a packet of encoded symbols S(k) and an encapsulation message comprising a field indicating the length of said packet;

b) discriminating said encapsulation message in order to generate, on the basis of the packet length, an envelope logic signal having a first binary value prior to the start and subsequently at the end of said packet of encoded symbols and a second binary value during the reception of said packet of encoded symbols;

c) subjecting said envelope logic signal and said encoded symbols to a logic processing enabling to generate:

successive pause symbols of specified value and rank, on the basis of said stable starting binary value, for said first binary value of said envelope logic signal, and

successive validated encoded symbols S'k={a'(k); b'(k)} of rank k corresponding to said symbols S(k) for said second binary value of said envelope logic signal; and

d) subjecting said successive validated pause symbols and validated encoded symbols of given rank to a continuous VITERBI type decoding, said pause symbols ~~enabling the~~ providing continuity of ~~the~~ a decoding lattice ~~to be obtained~~ by imposing a stable state between two packets of successive encoded symbols.

12. (Currently amended) A decoding method according to claim 11, wherein for a pause state as well as for a first logic value of said envelope logic signal corresponding to ~~the signaling of the packet~~ start of a packet and end of a packet equal to the value of zero and for a second logic value of said envelope logic signal equal to the value 1, said logic processing comprises applying to each received symbol a(k), b(k) an AND logic function with said envelope logic signal.

13. (Currently amended) A decoding method according to claim 11, wherein for a pause state as well as for a first logic value of said envelope logic signal corresponding to ~~the signaling of the packet~~ start of a packet and end of a packet equal to the value 1 and for a second logic value of said envelope logic signal equal to the value 0, said logic processing comprises applying to each received symbol a(k), b(k) an OR logic function with said complemented envelope logic signal.

14. (Currently amended) A decoding method according to claim 11, wherein for a pause state and a first logic value of said envelope logic signal equal to the value of 0, and for a signal representing ~~the~~ activity of the envelope corresponding to ~~the~~ a second logic value equal to the value 1, said logic processing comprises applying to each symbol a(k), b(k) an AND logic function with said complemented envelope logic signal.

15. (Currently amended) A decoding method according to claim 11, wherein for a pause state and a first logic value of said envelope logic signal equal to the value 1, and for a signal representing ~~the~~ activity of the envelope corresponding to the second

envelope value equal to 0, said logic processing comprises applying to each symbol a(k), b(k) an OR logic function with said envelope logic signal.

16. (Currently amended) A decoding device for a series flow of digital data encoded by convolutive encoding using a stable stating binary value and transmitted by packets, said device comprising:

means for receiving a packet of encoded symbols S(k) and an encapsulation message comprising a field indicating the length of the packet;

~~discriminating~~ means for discriminating said encapsulation message for enabling an envelope logic signal to be generated for said current packet, on the basis of ~~the~~ packet length, said envelope logic signal having a first ~~logic~~ binary value before the beginning and after the end of the encoded symbol packet and a second ~~logic~~ binary value during ~~the~~ receiving of the packet;

logic processing means for generating:

successive pause symbols of specified value and rank, on the basis of said stable starting binary value, for said first binary value of said envelope logic signal, and

successive validated encoded symbols S'k={a'(k); b'(k)} of rank k corresponding to said symbols S(k) for said second binary value of said envelope logic signal; and

VITERBI decoding means for carrying out, on the basis of said pause symbols and validated encoded symbols of given rank, a continuous VITERBI type decoding, said pause symbols ~~enabling the~~ providing continuity of ~~the~~ a decoding lattice to be obtained by imposing a stable state between two packets of successive encoded symbols.

17. (Currently amended) A device according to claim 16, wherein said logic processing means include at least:

a first logic cell receiving, in a first input, each first symbol element a(k), and, in a second input, said envelope logic signal, and delivering a first validated symbol element ~~s'~~a'(k); and

a second logic cell receiving, in a first input, each second symbol element b(k), and, in a second input, said envelope logic signal, and delivering a second validated symbol element b'(k).

18. (Currently amended) A device according to claim 17, wherein said first logic cell and said second logic cell comprise an identical logic gate, each gate having a first input and a second input, the first input of each logic gate constituting the first input of each logic cell and the second input of each logic gate being connected to the second input of each logic cell directly or by means of a logic inventer, as a function of the logic value assigned to the first and second binary value of said envelope logic signal, respectively.